

Secure and efficient data storage with Rivest Shamir Adleman algorithm in cloud environment

Ezhilarasan Elumalai¹, Dinakaran Muruganandam²

¹School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, India

²School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

Article Info

Article history:

Received Apr 7, 2023

Revised Nov 19, 2023

Accepted Feb 12, 2024

Keywords:

Cloud computing

Cloud storage server

Data security

Decryption

Encryption

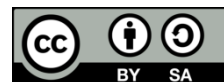
Secure and efficient data

Storage and retrieval algorithm

ABSTRACT

Cloud computing rapidly is a prerequisite and releases resources with minimal management effort. The surfacing of the cloud has significantly distorted the general insight into infrastructure, software services, and development models. In contrast to single-key encryption models based on public or private keys (PKs), hybrid encryption systems combine encryption methods using symmetric or asymmetric methods. Various hybrid algorithms fail to meet users' expectations regarding data security and cannot prevent all security risks. The secure and efficient data storage and retrieval (SEDSR) algorithm was developed for scalable key management between the content owner, cloud user, and service providers in an untrusted cloud environment. In the implementation, the SEDSR combines the Rivest Shamir Adleman (RSA) algorithm 4096 key length with a primary symmetric key method to provide adequate and compact security with optimal retrieval systems in the cloud. Based on the experimental evaluation, the SEDSR minimizes 1.7 seconds of encryption times (ET) and 1.5 seconds of decryption time (DT) and improves by 34% throughput (TRP) compared to existing parameters.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Dinakaran Muruganandam

School of Computer Science and Engineering, Vellore Institute of Technology

Chennai, Tamil Nadu, India

Email: dinakaran.m@vit.ac.in

1. INTRODUCTION

Cloud computing is a relatively emerging technology that provides users with on-demand network access to a shared computer resource that may be configured in various ways [1]. The surfacing of the cloud has significantly distorted the general insight into infrastructure, software delivery services, and development models [2]. It could merge many domains, such as networks, databases, operating systems, virtualization, resource scheduling, load balancing, concurrency control, network security, and transaction and memory management [3]. But both individual users and Industries require more privacy to deploy their applications or solutions in cloud environments.

Problem formulation: many cryptography techniques are available such as data encryption standard (DES), triple data encryption standard (TDES), Blowfish, Rivest Shamir Adleman (RSA), advanced encryption standard (AES), and elliptical curve cryptography (ECC) combining other algorithms to improve data security [4]. But many hybrid algorithms must meet user expectations for data security and prevent all security concerns and risks. Designing unique and robust security methods to increase digital data security is crucial as it is unavoidable in the digital era. However, these techniques could not be more helpful in maintaining the efficiency of key generation, encryption, and decryption. Current approaches still emphasize user identification, mutual privacy, and session-wise key agreement among content owners, trusted users, and

cloud service providers. But these systems still need to be improved by key complexity, malicious attack activity, and the difficulty of maintaining data continuity with the minimum setup.

Contribution: the secure and efficient data storage and retrieval (SEDSR) algorithm proposes to provide scalable key management among content owners, cloud users, and cloud service providers in the un-trusted cloud environment. The proposed method maintains the reliable credential verification process for cloud service providers. The proposed method combines the RSA algorithm and an essential symmetric key (SSK) method to provide adequate and compact security with optimal retrieval systems in the cloud.

The study aims to implement a symmetric-based encryption technique in a cloud environment to preserve owners' data privacy effectively and securely. Assume the cloud user wants to retrieve some required information. Two components need to be followed online one is cloud user request, and another is content owner data. In the database for data transfer in the cloud, each piece of data contains a unique ID.

First, the cloud user will fill out the registration form and search for the required information in the cloud. Now, the cloud users send the information search for the required information to get the data from the content owner in the cloud. Another component is the content request, which is compulsory to perform in the cloud to retrieve the information.

The SEDSR mechanism encrypts content owner and data, then communicates and sends encrypted messages to the cloud users. The users decrypt the encrypted data from the cloud storage server after secret key (SK) validations by using their private key (PK) and the user's public key. The proposed system improves encryption through RSA 4096 key length, offering greater security and privacy.

The rest of the article is organized as follows: section 2 discusses the proposed system's technique, the implemented module, and the algorithm execution steps in detail. Section 3 discusses the findings are explained with the specific deployment set, the simulation results outcomes, and comparative analysis. Section 4 summarizes with overall work of the study with future outcomes.

The approaches allowed users to securely transmit data inside a cloud system using encryption techniques such as RSA and Blowfish. The technologies allowed for flexible access to the computer systems, ultimately improving their cloud applications' performance. Anjana and Singh [5], discussed encryption of uploaded files in the cloud using double encryption techniques such as AES and RSA. Various approaches were presented in [6] to achieve high levels of data security. In this scenario, the hybrid approaches merged the ECC and Blowfish algorithms to provide high patient data protection while maintaining its secrecy. ECC is utilized for the encryption and decryption of data sent over the cloud because of the tiny size of the key used in ECC, which results in low computing power and, consequently, low energy consumption levels [7]. The RSA algorithm is used to support a PK cryptosystem, and it is used to handle both the user authentication system that is based on the blockchain (BC) and the improvised public-key cryptosystem [8]. Encrypting data sent across the cloud using cutting-edge cryptographic procedures was the primary emphasis of these methods in network security.

The TDES algorithm was developed to ensure the safety of massive data stored on the cloud. The TDES algorithm offers a substantially simpler solution that protects the data against attacks and maintains data privacy [9]. This is accomplished by raising the sizes of the keys used in the DES. By using the SHA-256 function in the cryptographic algorithms, these approaches improve security, verify the sharing of sensitive information, and authenticate the exchanged data [10]. Blowfish for data encryption and RSA for transferring Blowfish's SK [11] integrated various security approaches to provide a new lightweight security framework for authentication and data storage in the multi-cloud environment. Ghanmi *et al.* [12] derived reconfigurable multi-cloud storage server (MCSS) architecture for dynamic and secure data sharing has been designed. It concentrated on the MCSS and the data life cycle, which consists of three phases (i.e., data input, transition, and use) for measuring the effectiveness of reconfigurable data file slicing, standard format, privacy, and trustworthiness of the consumers. The technique is based on a BC-based architecture for logistics management that operates on a decentralized peer-to-peer network and uses Ethereum intelligent contracts [13]. Table 1 explains comparative studies of existing methods.

A cloud storage system was shown in [14] that used a hybrid cryptography approach. This method used the benefits of symmetric and asymmetric key cryptographic approaches to achieve optimal performance. Jian-Foo and Swee-Huay [15] developed, a method to be decentralized. It uses the AES and RSA algorithms to provide a framework that prohibits unauthorized entities from gaining access to the data or messages of users. The random key generation and decryption process may be sped up using eight prime numbers of modified RSA (EPNR) [16]. Hermawan *et al.* [17] addressed a method for managing confidentiality in cloud-based archives. This approach may enhance the safety of sensitive archive material without negatively impacting the effectiveness of archive search. When implemented, the method can provide various login stages and codes, such as one time password (OTP), to mitigate attacks [18].

Sarumi and Longe [19], offered a secure method for storing and retrieving data in the cloud was made available. This method is solely in the control of the data's owner. The approach eliminated data duplication by combining a deduplication algorithm with an algorithm for data integrity [20]. The data that is

looked at will be in an encrypted form, and the data from the hash will be utilized to check whether or not the data are accurate. The protocol is used for inter-UAV communication in a topology-changing network that uses a grid mobility model [21]. It is a method that relies on a pairwise comparison of several criteria to arrive at a prioritized ranking showing each choice alternative's overall effectiveness [22]. Each field of computational intelligence is examined in this research study from the cyber security perspective, along with its benefits and drawbacks [23]. It tackles the problem of making reliable recommendations for freshly installed cloud services/resources when more data is needed to support their reliability [24]. Wahab *et al.* [25] described the RSA algorithm in the cloud to ensure that data are kept safe and secure. RSA is encryption that prevents each message from being mapped to a number.

Table 1. Comparative studies of existing methods

S. No	Technique	Advantages	Disadvantages
1	Des	Des is a cryptography technique with a symmetric key block cipher that comprises two procedures, namely the procedure of encryption or encoding and the decoding.	The methods follow Egyptian circumstance that utilizes a single key for encoding and decoding.
2	3des	The data encryption standard (3des) is comparable to des but applies three times as many encryptions to the average safe time to protect the data.	Unlike other block cipher methods, 3des operates slower during the data encoding and decoding.
3	Rc5	The rc5 encryption method was created to provide an extraordinarily high level of privacy against several attacks, including brute-force attacks and differential cryptanalysis.	But, RC5 is vulnerable via side channels using timing and power analysis techniques.
4	Blowfish	The symmetric encryption method, Blowfish, uses a comparable SK to encrypt and decrypt the messages. It provides more privacy to the messages and gives high-end information privacy during information broadcasting in a risky medium.	The method generates a new 64-bit chunk by adding the 64-bit result to the blowfish cipher. Then, it duplicates each value in the p-array and each s-box in order.
5	AES	The block cipher model AES has a 128 bits block length. The encryption procedure includes 10 iterations of key processing using 128 bits, 12 iterations of key processing using 192 bits, and 14 iterations of key processing using 256 bits.	But, the number of columns in the AES method depends on the block's size for applying data privacy.
6	RSA	RSA algorithm was implemented from a public-key cryptosystem and digital signatures point of view. It includes public and PKs in data storage and data retrievals.	However, the method's available key length is incapable of maintaining robust privacy and is efficient for SK generation and validations.

2. SYSTEM METHOD

The proposed method described enhanced security during data sharing and information retrieval in an un-trusted cloud environment. Here, the content owner can browse the contributed file from their local computers and apply the proposed method to maintain strong security. After successful data encryption, a SK will be generated, and the data will be saved into the cloud storage server. Registered cloud users can log in and proceed with content retrieval. Once the system receives input from the user, the proposed system performs secure data sharing and retrieval with minimal time in the cloud. Hence, the retrieved file is extracted then methods ask the user to enter the valid S to identify the user [26]. The SEDSR algorithm is expressed in Figure 1.

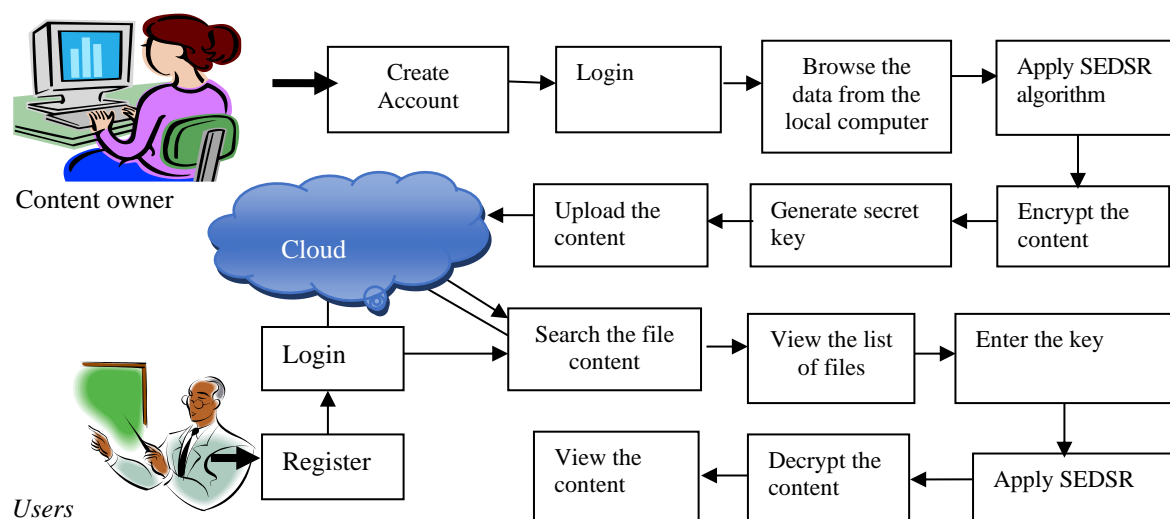


Figure 1. System architecture for SEDSR algorithm in the cloud

- Content owner: here, the content owner can register in the cloud vendor and will go for login after activating the owner account. Next, they can move to contribute the data in an un-trusted cloud environment. Here, the user can contribute unlimited data in multiple formats.
- Users: users will register here to access the data from a cloud environment. After login, the client can view the list file with the respective content's owner. However, the client is required authenticated key to access the full content.
- File uploading with encryption: in this process, the content owner can contribute their data from the local PC to the cloud after the data encryption. The proposed protocol generates a public and PK for the respective content during data encryption. Hence, the SK sends users to download the content based on request. This process ensures secure data transmission in the cloud.
- Data downloading with decryption: a client can securely view and download the data from an un-trusted cloud with a valid key. Users search the data to download the file and then use the SK that they have been given to decrypt the data that has been downloaded. Before the cloud user can decrypt the data, they must verify their permission to access the content. Hence, the user can access the downloaded data.

2.1. Secure and efficient data storage and retrieval algorithm with RSA algorithm

A SEDSR algorithm proposes to provide scalable key management among content owners, cloud users, and cloud service providers in the un-trusted cloud environment. The proposed method maintains the reliable credential verification process for cloud service providers. It provides a comprehensive framework to identify and prevent malicious activities in an un-trusted cloud environment. The proposed method is highly dedicated to avoiding fundamental complexities. When the data has been encrypted and a SK has been generated, the process continues with the data being moved to a cloud server then the content owner can store their data in the selected cloud. Cloud users and content owners can utilize the cloud services based on their subscription time. If somebody engages in any malicious activity, detecting and preventing it is possible. A valid user ID is required during the SK generation procedure to produce a SK that allows users to access the cloud by themselves. This method is only valid for approved users.

The proposed protocol is expressed as feasible in theory, where the computation is elaborated as an encoded combinational Boolean circuit that permits evaluating encrypted private data. The proposed technique takes only two rounds of communications for query retrieval. The method improves upon the RSA technique by adding a key length of 4096, which is more effective for producing SKs and their distribution, even in an insecure medium. The user's SK reflects the access mechanisms in this context. Therefore, the authorized user will only be able to verify the data transaction if and only if the characteristics of the data transaction satisfy the requirements for access to the secret key. The proposed method minimizes the key's encryption, decryption, and complexity in the solution model. The following procedure is included in the proposed design:

- Setup: the method reverses the PK based on the input parameter, a collection of integers, alphabets, and a unique character denoted by K. The production of a one-of-a-kind data transaction ID requires both a user secret key (USK) and PK.
- Secret key generation: the generation of two large random prime numbers, h and k , that have almost the same size so that the data $N=hk$ has the required bit length, for example, 4096 bits, is accomplished using this method. Calculating N as the SK exponent SE, $1 < SE < \phi$ such that $iSE \equiv 1 \pmod{\phi}$. The public key is written as (N, I) , and the PK is (SE, h, k) . It secretly keeps track of SE, h , k , and ϕ values. Sometimes, the PK is written as per the required value of N when using SE. Otherwise, it may express the key pair as $((N, i) SE)$, where N is referred to as the modulus and i as the public or encryption exponent. To function, the method takes the inputs h and k for the access tree structure, as well as the USK. The approach offers a SK or signature S , enabling users to verify the data transactions for respective users. Only a user who has been given the appropriate permissions may generate USK.
- Secret key validations: users can make some modifications, such as producing a message digest of the data that will be delivered. It presents the message or content digest as an exponent DE that ranges between 1 and $N-1$. The method uses a PK (N, SE) to compute the signature $S = DE^{SE} \pmod{N}$. It transmits this signature S to the receiver. The sender uses the public key exponent (N, i) to calculate the integer $x = S^i \pmod{N}$ on the receiver side. It estimates the message or content digest of the data that has been signed independently. It computes the expected representative integer x' by encrypting the expected message digest if $x = x'$, then the signature is correct. The method receives input from users with their signature S to perform the data transaction validation process and key access structure.
- Encryption: get a hold of the recipient's public key and then represent the plaintext content as a positive integer PI with $1 < PI < N$. The method estimated the cipher text $CT = PI^i \pmod{N}$ and sent it to the recipient.

- Decryption: the receiver uses their PK (N, SE) to compute $PI = CTSE \bmod N$ depending on the size of SE and N. This calculation is based on the size of SE and N. It takes the message or content representing PI and extracts the plain text from them.

The proposed algorithm's pseudo code is shown in detail:

Input: Take input as a set of integers, alphabets, and special characters K
Output: UUID-based SK generation, encryption time (ET), decryption time (DT), throughput (TRP)

Procedure:
Start
Start the procedure;
Enter the necessary attributes for the secure transactions process in a cloud;
Proceed for Enhanced UUID-based SK generations;
Take the parameters K;
Proceed for computations;
If the computation process is completed, then
 SK generated from access structure A;
 Proceed for secure transaction and data uploading process is completed in a cloud;
 Initialize the data decryption and downloading process;
 Enter the secret SK for validation of the right users;
If SK is validated, then
 The data decryption process is completed, and SK is taken from access structure A;
Else
 Unauthorized users and process is declined;
Else
 Restart the process;
End

3. RESULT AND DISCUSSION

3.1. Implementation setup

The implementation is done on a desktop computer with a Windows 10 Professional operating system, an Intel i3 Core CPU processor, 8 GB RAM, and 500 GB memory. The development has been deployed on NetBeans 8.0.2, JDK 1.8, Apache Tomcat 8.0.27, MYSQL 8.0, and the cloud environment provided by Jelastatic. The proposed protocol is evaluated multiple kinds of content (0.5 MB and 1 MB) along the closest existing approaches.

3.2. Simulation result

This part presents a mathematical derivation to evaluate the efficiency of the data transformation, security, processing speed, and TRP. The SEDSR performs its duties on the side of the content owner and the cloud user in an untrustable cloud environment. The proposed method is only partially trusted on cloud storage servers for data sharing and retrieval in a secure manner. Hence, the proposed method of SEDSR algorithms evaluates ET, DT, and TRP on different data sizes.

3.2.1. Encryption time

The proposed method derives a mathematical model for the ET and is expressed in (1). The inputs that are taken into consideration by this method are the content or message M, the public key PK, and the attribute I. The following is how the cipher-content CC operation is carried out by it:

$$CC = (I, CC\{CC_i\}_{i \in I}) \quad (1)$$

Where $\sim CC$ represents MY^s , CC_i denotes A_i^s , and s is randomly chosen from Z_p .

3.2.2. Decryption time

The proposed method shows a mathematical model of DT in (2). This function inputs a cipher content CC encrypted using the attribute set I. The method takes as input the user's SK for access tree A and the public key PK. It shows the mathematical expression as (2):

$$CC(CC_i, ski) = CC(g, g)^{pi(0)s} \quad (2)$$

Where CC=chipper content, SK_i =user SK component for attribute I for leaf nodes. Then, it combines the pair result in sequential order. Finally, it enhances the sightless factor $Y^s = CC(g, g)$ and produces C content if I satisfy A.

3.2.3. Throughput

The TRP is a successful data transmission and retrieval time with data size in untrusted cloud environments. The proposed system is estimated as the TRP of the mathematical model in (3). The following formula is used to determine TRP:

$$\text{Throughput} = \left(\frac{\text{PlaintText}_{\text{Size}}}{\text{Encryption}_{\text{Time}}} + \frac{\text{ChiperText}_{\text{size}}}{\text{Decryption}_{\text{Time}}} \right) \times 100 \quad (3)$$

$\text{PlaintText}_{\text{Size}}$ is the total amount of encrypted data and $\text{ChiperText}_{\text{size}}$ is the decrypted data. The $\text{Encryption}_{\text{Time}}$ and $\text{Decryption}_{\text{Time}}$ represent the overall times to process the encryption and decryptions. Here, TRP put is measured on a scale of 100.

Table 2 is expressed the ET, DT, and TRP of the proposed SEDSR+RSA 4096 with the closest existing approaches. SEDSR is evaluated with closed conventional methodologies, namely DES, Blowfish, RC5, 3-DES and AES+RSA. The method is studied in terms of ET (second), DT (seconds), and TRP in %, presenting their average values for corresponding parameters with a variety of different sizes of data. Where it was observed that the AES+RSA hybrids method is the closest competitor to our proposed methods, but the number of columns in the AES method depends on the block's size, and the RSA method's existing key length cannot maintain strong privacy or upgrade cloud environments. Based on tabular results, the proposed SEDSR+RSA 4096 method performs well on every respective parameter compared to the closest methods.

Table 2. ET, DT, and TRP for 0.5 MB and 1.0 MB data

Learning algorithm	0.5			1		
	ET	DT	TRP	ET	DT	TRP
DES	3	2.5	20	5.5	4	21
Blowfish	1.6	1.8	29	4.5	3.5	25
RC5	1.8	2	35	4.8	3.5	24
3-DES	2.5	2.3	21	5	3.8	23
AES+RSA	1.5	1.5	33	4	3	28
SEDSR+RSA 4096	0.8	1.0	55	3	2	40

The ET, DT, and TRP are evaluated with our closest approach, AES+RSA hybrid techniques. AES is a block cipher model, and the encryption procedure includes 10 iterations of key processing using 128 bits, 12 iterations of key processing using 192 bits, and 14 iterations of key processing using 256 bits. But, the number of columns in the AES method depends on the block's size for applying data privacy. RSA algorithm was implemented from a public-key cryptosystem and digital signatures point of view. It includes public and PKs in data storage and data retrievals. However, the method's general key length could be more capable of maintaining robust privacy and efficient for SK generation and validations. An improvement in DES is known as 3DES. The 3DES is comparable to DES but applies three times as many encryptions to the average safe time to protect the data. However, unlike other block cipher methods, 3DES operates slower during the data encoding and decoding. The RC5 encryption algorithm was developed to provide an exceptionally high degree of privacy against several attacks, such as brute-force attacks and differential cryptanalysis. But, RC5 is vulnerable via side channels using timing and power analysis techniques. Blowfish is a symmetric encryption technique that utilizes a similar SK to encode and decode messages. DES is a cryptography technique with a symmetric key block cipher that comprises two procedures, namely the procedure of encryption or encoding and the decryption or decoding. DES encryption results in cipher content of 64 bits, and decryption is achieved by reversing the encryption or encoding process stages. However, the methods follow Egyptian circumstance that utilizes a single key for encoding and decoding. Based on the graphical result of Figures 2 to 4, proposed SEDSR+RSA 4096 displays the result of ET, DT, and TRP for 0.5 MB and 1.0 MB data size.

The proposed method maintains the reliable credential verification process for cloud service providers. It provides a comprehensive framework to identify and prevent malicious activities in an un-trusted cloud environment. The proposed method is highly dedicated to avoiding fundamental complexities. When the data has been encrypted and a SK has been generated, the process continues with the data being moved to a cloud server then the content owner can store their data in the selected cloud. The proposed technique takes only two rounds of communications for query retrieval. The method improves upon the RSA technique by adding a key length of 4096, which is more effective for producing secret keys and their distribution, even in an insecure medium. The user's SK reflects the access mechanisms in this context. Here, the proposed SEDSR+RSA 4096 method reduces 1.7 seconds ETs and 1.5 seconds DT and improves by 34% TRP

compared to our conventional methodologies. Based on Tabular and graphical results, it has been observed that the proposed SEDSR+RSA 4096 method performs well on every respective parameter compared to the closest methods.

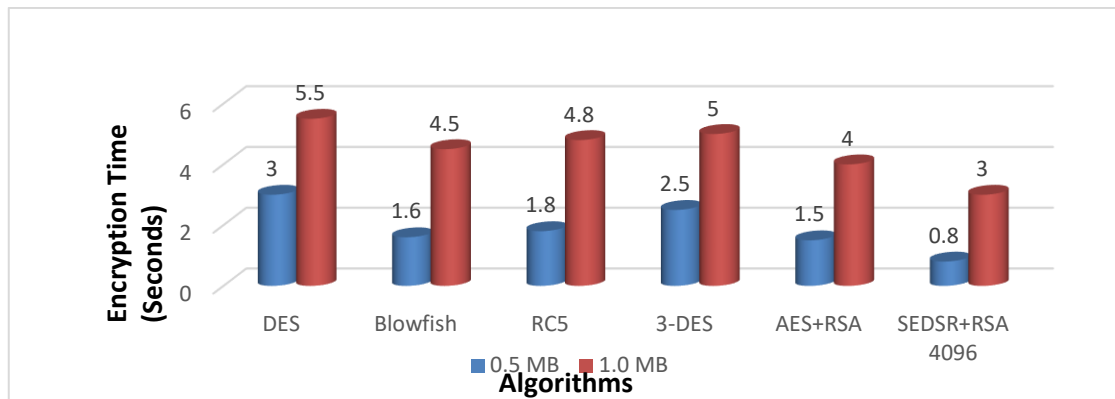


Figure 2. ET for 0.5 MB and 1.0 MB data

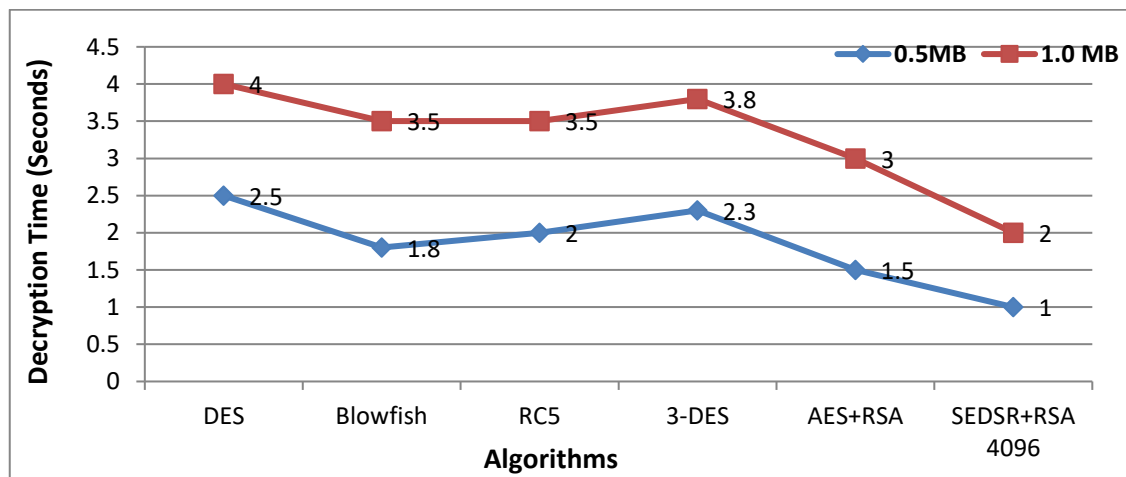


Figure 3. DT for 0.5 MB and 1.0 MB data

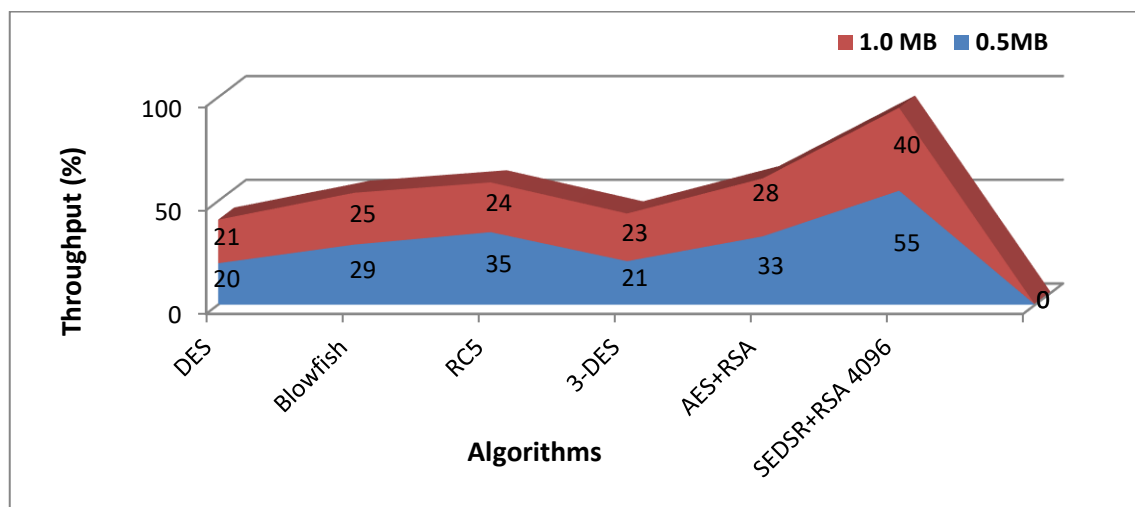


Figure 4. TRP for 0.5 MB and 1.0 MB data

4. CONCLUSION

The SEDSR algorithm is designed for scalable key management between the content owner, cloud users, and cloud service providers in an un-trusted cloud. The proposed method combined the RSA algorithm with the primary symmetric key method to provide adequate and compact security with optimal retrieval systems in the cloud. The proposed system improves encryption through RSA and a recently announced symmetric algorithm to offer greater security and privacy. The proposed approach mainly focused on avoiding key complexity. When the data has been encrypted and the SK has been generated, the operation will move to the cloud server for storing the content owner's data in his selected cloud server. Cloud users and content owners may access the cloud storage based on their subscription time. A valid user ID is required during the SK creation procedure to produce a key allowing users to access the cloud themselves. Only authorized users are allowed to use this procedure. Including a key length 4096 enhances the RSA methodology and is more efficient for creating secret keys and distributing them even through insecure channels. The access techniques are reflected in the user's secret key. Here, the proposed SEDSR+RSA 4096 method reduces 1.7 seconds ETs and 1.5 seconds DT and improves by 34% TRP compared to our conventional methodologies. Based on Tabular and graphical results, it has been observed that the proposed SEDSR+RSA 4096 method performs well on every respective parameter compared to the closest methods.

The research paper can be improved so that the proposed algorithm will be enhanced and deployed in BC technologies with the required features for strong token generation and certificate validations. It will be designed in upgraded environments to utilize cloud-based real-time products/applications. It needed more effort to bring optimal solutions for a larger volume of sensitive data of various types.




REFERENCES

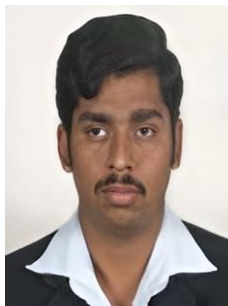
- [1] N. Ghazy, A. Abdelkader, M. S. Zaki, and K. A. Eldahshan, "An ameliorated Round Robin algorithm in the cloud computing for task scheduling," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 1103-1114, 2023, doi: 10.11591/eei.v12i2.4524.
- [2] M. S. Kumar, H. Azath, A. K. Velmurugan, K. Padmanaban, and M. Subbiah, "Prediction of Alzheimer's disease using hybrid machine learning technique," *AIP Conference Proceedings*, vol. 2523, pp. 1-6, 2023, doi: 10.1063/5.0110283.
- [3] A. Unnikrishnan and V. Das, "Cooperative Routing for Improving the Lifetime of Wireless Ad-Hoc Networks," *International Journal of Advances in Signal and Image Sciences*, vol. 8, no. 1, pp. 17-24, 2022, doi: 10.29284/ijasis.8.1.2022.17-24.
- [4] F. Sajid *et al.*, "Secure and efficient data storage operations using intelligent classification techniques and RSA algorithm in IoT-based cloud computing," *Scientific Programming*, vol. 2022, pp. 1-10, 2022, doi: 10.1155/2022/2195646.
- [5] Anjana and A. Singh, "Hybrid Cryptographic solution using RSA, Blowfish, and MD5 for Information Security in Cloud Computing," *Mathematical Statistician and Engineering Applications*, vol. 71, no. 3s, pp. 1250-1268, 2022.
- [6] K. Jaspin, S. Selvan, S. Sahana, and G. Thanmai, "Efficient and secure file transfer in the cloud through double encryption using AES and RSA algorithms," In *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pp. 791-796, 2021, doi: 10.1109/ESCI50559.2021.9397005.
- [7] P. Chinnasamy, S. Padmavathi, R. Swathy, and S. Rakesh, "Efficient data security using hybrid cryptography on cloud computing," In *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020*, Springer Singapore, pp. 537-547, 2021, doi: 10.1007/978-981-15-7345-3_46.
- [8] G. O. Ogunleye and S. E. Akinsanya, "Elliptic Curve Cryptography Performance Evaluation for Securing Multi-Factor Systems in a Cloud Computing Environment," *Iraqi Journal of Science*, vol. 63, no. 7, pp. 3212-3224, 2022, doi: 10.24996/ijis.2022.63.7.40.
- [9] I. A. A. Samy and M. S. Mary, "Secure Data Transmission in Cloud Computing Using Std-RSA with Eslurnn Data Classification and Blockchain-Based User Authentication System," *Research Square (preprint)*, pp. 1-21, 2022, doi: 10.21203/rs.3.rs-1724672/v1.
- [10] M. N. Ramachandra, M. S. Rao, W. C. Lai, B. D. Parameshachari, J. A. Babu, and K. L. Hemalatha, "An Efficient and Secure Big Data Storage in Cloud Environment by Using Triple Data Encryption Standard," *Big Data and Cognitive Computing*, vol. 6, no. 101, pp. 1-20, 2022, doi: 10.3390/bdcc6040101.
- [11] E. A. Adeniyi, P. B. Falola, M. S. Maashi, M. Aljebreen, and S. Bharany, "Secure sensitive data sharing using RSA and ElGamal cryptographic algorithms with hash functions," *Information*, vol. 13, no. 442, pp. 1-15, 2022, doi: 10.3390/info13100442.
- [12] H. Ghanmi, N. Hajlaoui, H. Touati, M. Haddad, and P. Muhlethaler, "A Secure Data Storage in Multi-cloud Architecture Using Blowfish Encryption Algorithm," In *Advanced Information Networking and Applications: Proceedings of the 36th International Conference on Advanced Information Networking and Applications (AINA-2022)*, Cham: Springer International Publishing, vol. 2, pp. 398-408, 2022, doi: 10.1007/978-3-030-99587-4_34.
- [13] P. M. B. Muddumadappa, S. D. K. Anjanappa, and M. Srikantaswamy, "An efficient reconfigurable cryptographic model for dynamic and secure unstructured data sharing in multi-cloud storage server," *Journal of Intelligent Systems and Control*, vol. 1, no. 1, pp. 68-78, 2022, doi: 10.56578/jisc010107.
- [14] N. A. Ugochukwu, S. B. Goyal, A. S. Rajawat, S. M. N. Islam, J. He, and M. Aslam, "An Innovative Blockchain-Based Secured Logistics Management Architecture: Utilizing an RSA Asymmetric Encryption Method," *Mathematics*, vol. 10, no. 24, pp. 1-18, 2022, doi: 10.3390/math10244670.
- [15] L. Jian-Foo and H. Swee-Huay, "Secure File Storage on Cloud Using Hybrid Cryptography," *Journal of Informatics and Web Engineering*, vol. 1, no. 2, pp. 1-18, 2022, doi: 10.33093/jiwe.2022.1.2.1.
- [16] H. Jashn, B. Mahipour, E. Moharamkhani, and B. Zadmehr, "A Framework for Privacy and Security on Social Networks Using Encryption Algorithms," *International Journal of Smart Electrical Engineering*, vol. 12, no. 01, pp. 31-41, 2023.
- [17] N. T. E. Hermawan, E. Winarko, and A. Ashari, "Eight Prime Numbers of Modified RSA Algorithm Method for More Secure Single Board Computer Implementation," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 11, no. 6, pp. 2375-2384, 2021.




- [18] J. Xie, S. Xuan, W. You, Z. Wu, and H. Chen, "An Effective Model of Confidentiality Management of Digital Archives in a Cloud Environment," *Electronics*, vol. 11, no. 2831, pp. 1–17, 2022, doi: 10.3390/electronics11182831.
- [19] J. A. Sarumi and O. B. Longe, "Towards Data Storage Security in Cloud Computing Using Hybridized Advanced Encryption Standard & Authentication Scheme," *Journal of Digital Innovations & Contemporary Research in Science, Engineering & Technology*, vol. 10 no. 1, pp. 25–48, 2022, doi: 10.22624/AIMS/DIGITAL/V10N2P5.
- [20] M. S. Vedita and S. P. S. Naaz, "Secure File Storage on Cloud Using Elliptic Curve Cryptography (ECC) Algorithm," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 05, no. 01, pp. 1182–1184, 2023, doi: 10.56726/IRJMETS33137.
- [21] D. H. B. Sowri and S. D. Reshma, "Client-End Deduplication on Encrypted Data with Public Auditing in Cloud Storage," *International Journal of Techno-Engineering (IJTE)*, vol. 11, no. 01, pp. 39–47, 2019.
- [22] M. A. Hassan, A. R. Javed, T. Hassan, S. S. Band, R. Sitharthan, and M. Rizwan, "Reinforcing Communication on the Internet of Aerial Vehicles," in *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1288–1297, Sep. 2022, doi: 10.1109/TGCN.2022.3157591.
- [23] I. Kouatli, "People-process-performance benchmarking technique in cloud computing environment: An AHP approach," *International Journal of Productivity and Performance Management*, vol. 69, no. 9, pp. 1955–1972, 2019, doi: 10.1108/IJPPM-04-2017-0083.
- [24] M. A. Hassan, M. I. S. Ali, and B. Shaista, "New Advancements in Cybersecurity: A Comprehensive Survey," *Big Data Analytics and Computational Intelligence for Cybersecurity*, vol. 111, pp. 3–17, 2022, doi: 10.1007/978-3-031-05752-6_1.
- [25] O. A. Wahab, C. R. Cohen, B. Jamal, O. Hadi, M. Azzam, and R. Gaith, "An endorsement-based trust bootstrapping approach for newcomer cloud services," *Information Sciences*, vol. 527, pp. 159–175, Jul. 2020, doi: 10.1016/j.ins.2020.03.102.
- [26] A. P. M. Meenakumari, S. L., R. N., S. Jayaprakash, and S. Murugan, "Intelligent Power Control Models for the IOT Wearable Devices in BAN Networks," *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, Bengaluru, India, 2023, pp. 820–824, doi: 10.1109/IITCEE57236.2023.10090918.

BIOGRAPHIES OF AUTHORS



Ezhilarasan Elumalai    is a research scholar from the School of Information Technology and Engineering at Vellore Institute of Technology, Vellore, India. He has received a Master's in Computer Science Engineering at SCSVMV University in Kanchipuram, India. He received his UG in Computer Science Engineering at Anna University Chennai, India. He has 4+ years of teaching and industrial experience in various organizations in Tamil Nadu, India. He published 5 articles in major indexing (Scopus and Web of Science) journals. He participated in many international/national conferences, workshop seminars, and guest lecturers at various institutions in India. His research interests are cloud security, mobile computing, and mobile networks. He can be contacted at email: e.ezhilarasan2016@vitstudent.ac.in.



Dinakaran Muruganandam    is working as a Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India. He received his Ph.D. award from Anna University Chennai and a UG and PG degrees from Vellore Institute of Technology, Vellore, India. He has over 14 years of academic experience in various institutions. He published 53 articles in significant indexing (Scopus and Web of Science) journals. He participated in many international/national Conferences, workshop seminars, and guest lecturers at various institutions in India. His research interests are mobile computing, cloud computing, mobile networks, mobile IP, IPv6, and mobile telephony. He can be contacted at email: dinakaran.m@vit.ac.in.